Cont A1

is available when needed by the STB.

Page 5, line 18+:

A2

The most commonly filtered field within the section table layer syntax of MPEG-2 compliant data is the eight bit Table ID field since the table ID field is the most basic identifier for table sections and a hardware filter may be configured to send to memory all table sections having a particular PID value that also have a table ID field matching or partially matching a defined value. This is also an example of positive filtering where a match of one or more bits must be found for the data to be passed for storage. Such positive filtering with an effectively variable filter length provided by the above-incorporated application meets not only the applications program interface (API) definition of common middleware (e.g. between a low level operating system and an application program) implementations but extends the API to allow many programmable variations such as variable length filtering which can be critical for future applications such as internet protocol (IP) packet transmissions.

Page 6, line 25+:

A3

Unfortunately, while the filtering arrangement of the above-incorporated application would accommodate long runs of data, it is directed to only positive filtering. Prior to the present invention, no alternative exists for negative filtering of long runs of data other than registers and gate arrays sufficiently extensive to accommodate the maximum data bit string to be compared. Such an approach would require extensive hardware or present a substantial processing burden if implemented in software; neither of which is economically acceptable in a STB, particularly where a large plurality of filters must be implemented in separate hardware (to reduce overall

Cont A3

set-up time) as is generally desirable in STBs. The absence of such filtering action results in the storage of data which must be later processed/parsed and discarded. Since no alternative exists, however, unconditional storage of such data increases required storage capacity in the STB and imposes a processing burden that cannot be avoided.

Page 13, line 22+:

A4

The invention also provides a further not match indicator register 40 in a manner which is linkable to other filters and blocks and which is preferably four bits long to correspond to the four bytes of the mask, filter and data registers. The respective bits of the not match indicator register control whether positive or negative filtering is to be applied to the respective byte of data. Under the MPEG-2 standard, no greater granularity of mixing of positive and negative filtering is anticipated but a bit of the not match indicator could be provided for each bit or arbitrary group of bits of the mask, filter and data registers if desired. Thus, for example, if a not match bit NMn is "0", positive filtering will be applied to the corresponding byte n (or bit or group of bits) and if the not match bit is "1", negative filtering will be applied.

Page 16, line 18+:

A5

A filter ID is preferably implemented with a section filter ID and a pointer in the form of a "next filter ID" which correlates the section filter ID with the current block of data being filtered to supply appropriate mask and filter register data (e.g. along a row of Figure 4). The section filter ID and next filter ID are preferably specified in a control word along with other data such as a next column flag, a match/not match flag (which cannot be used to express

Cont A5

mixed filtering) or be correlated with mixed or negative filtering for another block to control arbitrary filtering over long bit strings.

Page 17, line 19+:

A6

Figure 3 similarly illustrates operation of the filter arrangement of the invention in regard to a next or subsequent block of data. For purposes of this discussion, this next or subsequent data block will be considered to contain byte 6, byte 7 byte 8 and byte 9. Again, the filter blocks are set up into a second column 310 (column 1) of M filter blocks (wherein M can but need not equal N and results in a thirty-two bit matchword MW1 which may immediately be combined with matchword MW0, in a manner depending on whether or not the current filter algorithm includes negative filtering. Specifically, the respective compare result CR bits of each matchword, as developed, is logically combined with the logically combined result of all prior matchwords using AND logic if the current filter block is a positive filter (NM = "0") and using OR logic if the current filter block is a negative filter.

Page 18, line 6+:

A7

Thus a single matchword will be developed at the end of each column and at the end of each variable length word, when all desired filter functions have been performed on all data blocks by all filter columns, as depicted in Figure 4. Thus each row of Figure 4 will correspond to a filter ID of arbitrary length. The depiction in figure 4 provides three arbitrary filter functions of at least ninety-six bits in length (three columns x data block length). Positive and negative filtering is tracked through the columns (in order to provide positive, negative or mixed filtering on each byte of each data block by the representation of matches under positive filtering and

Cont A7

mismatches under negative filtering by the same logic state of the corresponding CR value in the matchword formed over a column and accumulating the final matchword by respectively ANDing or ORing the respective CR values depending on whether the current filter block included positive filtering (e.g. positive or mixed filtering over the data block corresponding to an AND function but exclusively negative filtering over a block corresponding to an OR function for proper accumulating of the matchword although other matchword accumulation may be useful).

Page 19, line 12+:

A8

It should be recalled that full flexibility of filter function and variability of length of the datastream upon which filtering is performed is a meritorious effect of the invention. It should also be understood that, in practice, filter functions may be provided in memory (e.g. read only memory) of the STB or through downloading of a filter function from the transmitted data. A modification of a filter function in memory can also be downloaded from transmitted data. For these reasons, in practice, it is desirable to increase flexibility of filter function by providing for an extra SetMW bit 45 in the filter block 100, preferably in not match indicator register 40, as shown in Figure 1. This bit may be used to directly control use of an AND or OR function as the matchword is accumulated rather than evaluating the contents of the no match indicator register 40 as described above. Therefore, for example, if the length of the filter function were four blocks (sixteen bytes) long and the respective SetMW values for the four words (e.g. A, B, C, D, in the chronology of the filtering process) were "0110" the matchword would be accumulated in accordance with the expression (((A * B) + C) + D).